

FIG. 1

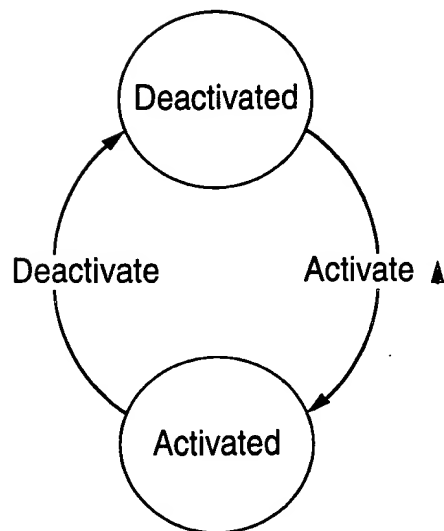


FIG. 2a

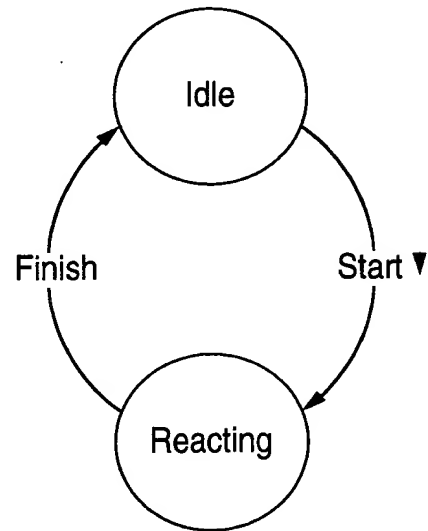


FIG. 2b

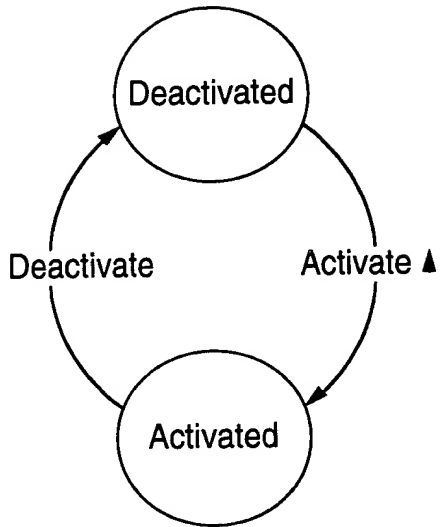


FIG. 3a

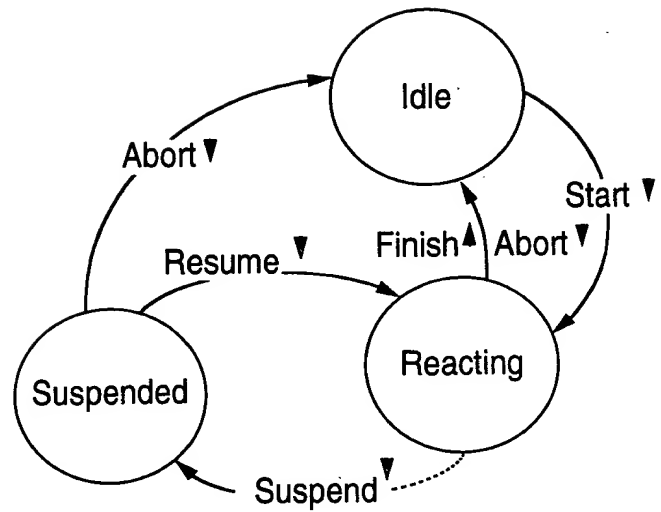


FIG. 3b

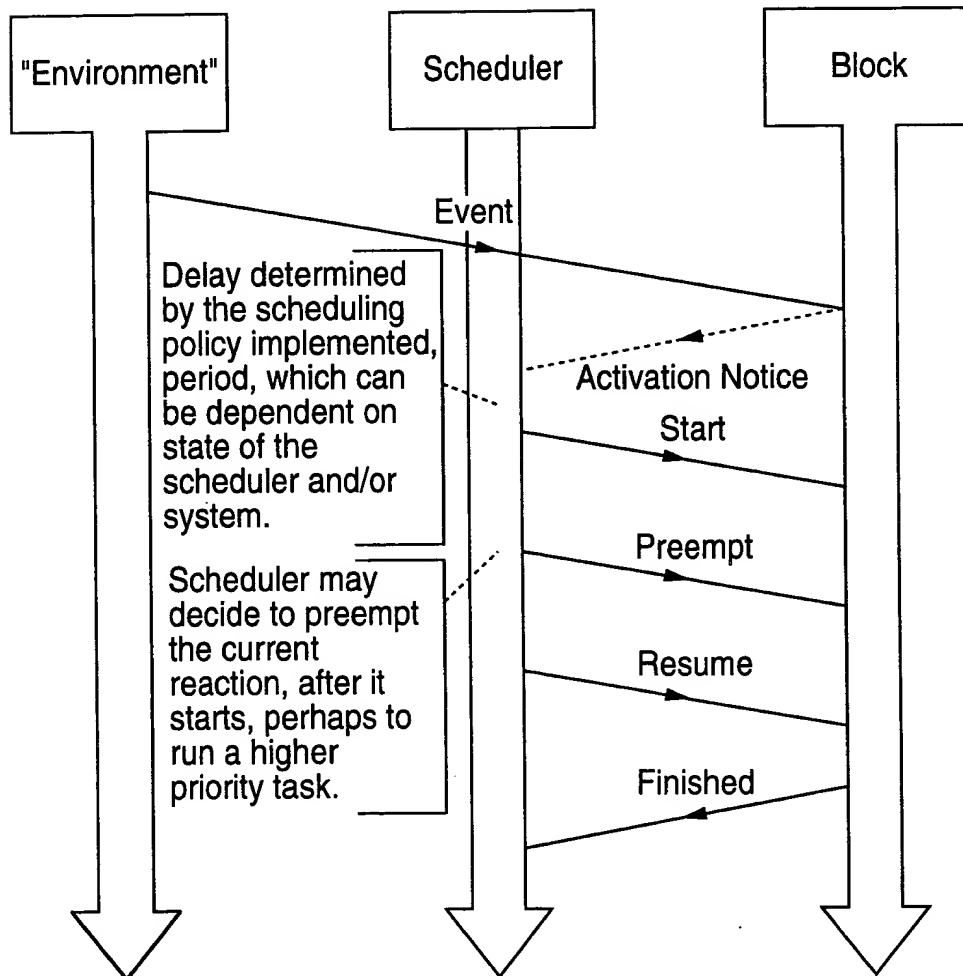


FIG. 4



3/5

```
class FxSchedulerinterface
{
public:
    //
    // Messages sent from FxSchedulableinterface objects.
    virtual void schedulerActivationNotice(
        FxSchedulableInterface *) = 0;
    virtual void schedulerFinishNotice(
        FxSchedulableInterface *) = 0;
};
```

FIG. 5

```
class FxSchedulableinterface
{
public:
    //
    // Messages sent to FxSchedulerInterface objects.
    virtual void schedulableStart() = 0;
    virtual void schedulableSuspend() = 0;
    virtual void schedulableResume() = 0;
    virtual void schedulableAbort() = 0;
};
```

FIG. 6

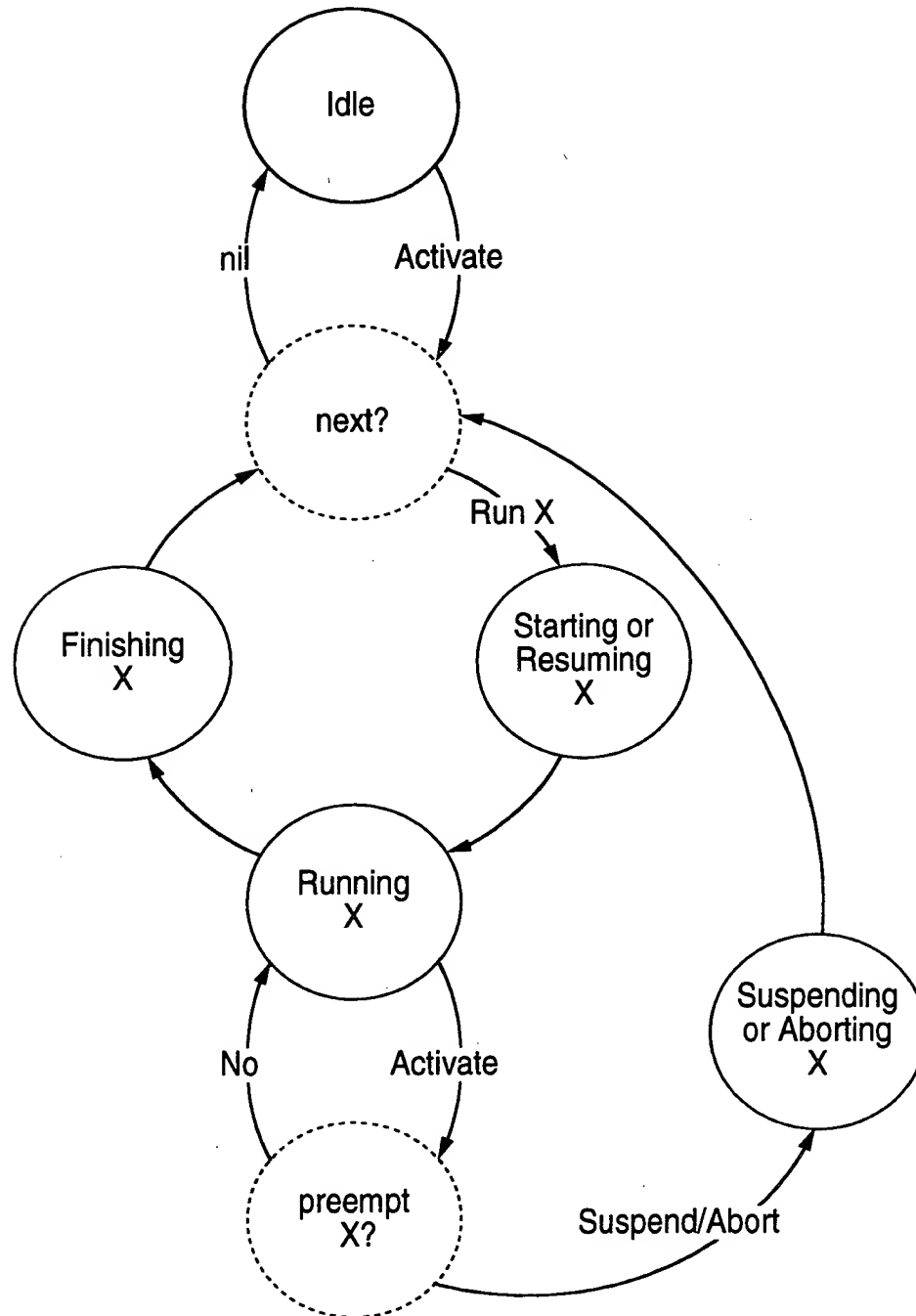


FIG. 7



```
task_ABD() (  
    while (1) ( // cyclo-static schedule in task  
        read_events();  
        A_Run(); // 1  
        B_Run(); // 2  
        D_Run(); // 3  
        post_events();  
    )  
)  
  
irq_handler_C() (  
    C_Run(); // interrupt handler  
    post_events();  
)  
  
task_E() (  
    while (1) ( // single behavior in task  
        read_events();  
        E_Run();  
        post_events();  
    )  
)  
  
main() (  
    // initialization  
    rtos_create_task(task_ABD, ...)  
    rtos_install_handler(irq_handler_C, ...)  
    rtos_create_task(task_E, ...);  
    ...  
)
```

FIG. 8